Regression tutorial

This is a quick tutorial with an example dataset on regressions in R. With this tutorial, you will be able to run linear regression models, get r-squared and p-values, and make appropriate figures. If you have any questions, please get in touch with me soon.

First step is to get your .csv file into Rstudio and make sure it looks right. (Also make sure you set your working directory to the location where your data file is). Your data may have treatment information, but it may also simply be X and Y values, depending on your projects.

```
### Set your working directory ###

Reg_data <- read.csv(file="Regression_data.csv")

head(Reg_data)
```

```
##   Treatment X_variable Y_variable
## 1   Control          2       10.0
## 2   Control          4       12.0
## 3   Control          3       12.0
## 4   Control          2       10.0
## 5   Control          5       13.0
## 6   Control          1       11.7
```

Next, run a linear regression to test whether X significantly correlates to Y. Remember, correlation does not always equal causation, so interpret the relationship cautiously unless you are certain that the direction is X affects Y, and not Y affects X. The summary output includes the significance test (p-value), and the strength of the correlation (adjusted R-squared).

```
Reg_model <- lm(Y_variable ~ X_variable, data = Reg_data)   # This code says to make an object based on

summary(Reg_model)
```

```
## 
## Call:
## lm(formula = Y_variable ~ X_variable, data = Reg_data)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -3.3738 -2.3194  0.5411  1.3550  4.1902 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  14.7459     1.3689   10.77 2.81e-09 ***
## X_variable   -0.7446     0.3339   -2.23   0.0387 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2.318 on 18 degrees of freedom
## Multiple R-squared:  0.2165, Adjusted R-squared:  0.173 
## F-statistic: 4.975 on 1 and 18 DF,  p-value: 0.03869
```

Let's say your data has an added treatment involved (like burned/unburned). You should run an "analysis of co-variance" or ANCOVA. This is a model that tests: 1) the effect of the X-axis on the Y-axis, 2) the effect of the treatment overall, and 3) if the effect of the X-axis on the Y-axis depends on the treatments. It takes one minor adjustment to the code to run an ANCOVA:

```
ANCOVA_model <- lm(Y_variable ~ X_variable * Treatment, data = Reg_data)

summary(ANCOVA_model)
```

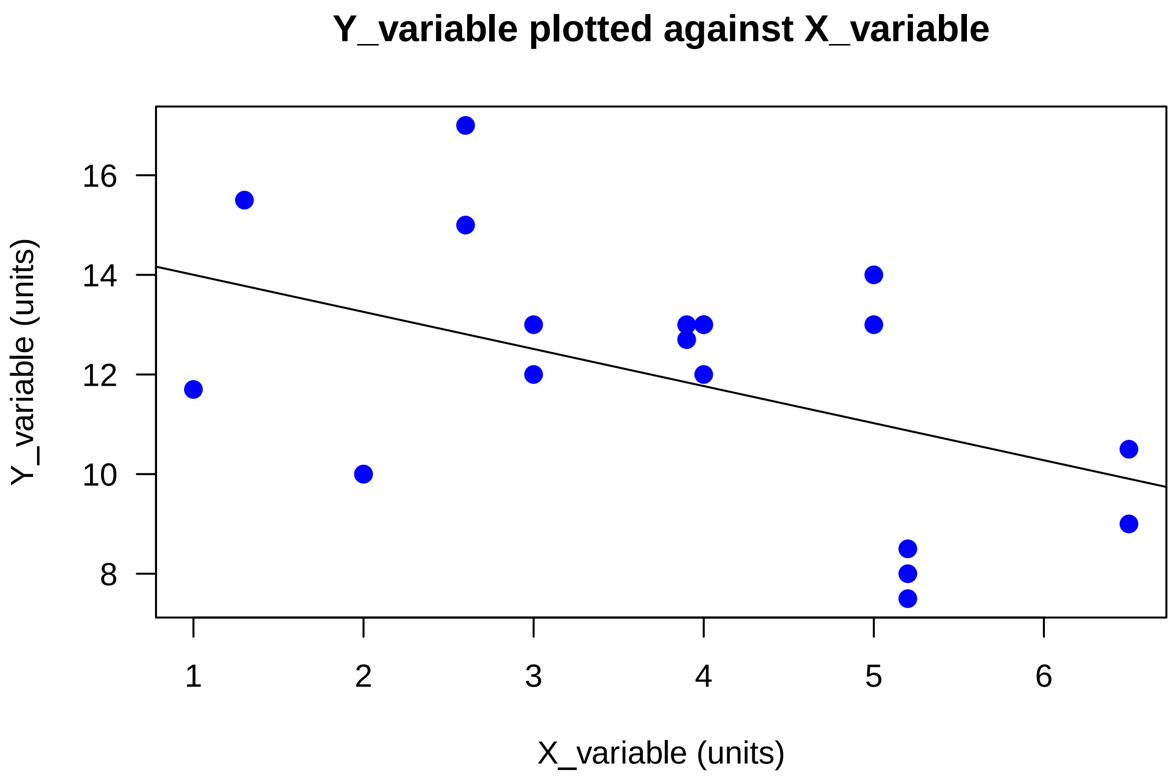
```
##
## Call:
## lm(formula = Y_variable ~ X_variable * Treatment, data = Reg_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6478 -1.1770  0.3044  0.7649  2.5267
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                          9.6919     1.3398   7.234    2e-06 ***
## X_variable                           0.7509     0.3790   1.982 0.064976 .
## TreatmentManipulation                9.1540     1.8948   4.831 0.000184 ***
## X_variable:TreatmentManipulation  -2.4237     0.4781  -5.069 0.000114 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.521 on 16 degrees of freedom
## Multiple R-squared:  0.7003, Adjusted R-squared:  0.6442
## F-statistic: 12.47 on 3 and 16 DF,  p-value: 0.0001858
```

Here, the R-squared refers to the overall variation explained by the three factors (X, Treatment, and X by treatments). If you want R-sqaured for invidual treatments, you can split your data (see other R tutorials for P1 and P2), or manually split your data in excel and read in 2 individual data files.

Now you want to make the regression figure. Code for a simple regression figure looks like this:

```
# First, make individual objects for your X and Y axis
X_variable <- Reg_data$X_variable
Y_variable <- Reg_data$Y_variable

# Run both these line of code together
plot(X_variable, Y_variable, pch = 16, cex = 1.3, col = "blue", las = 1, main = "Y_variable plotted aga
abline(lm(Y_variable ~ X_variable))
```



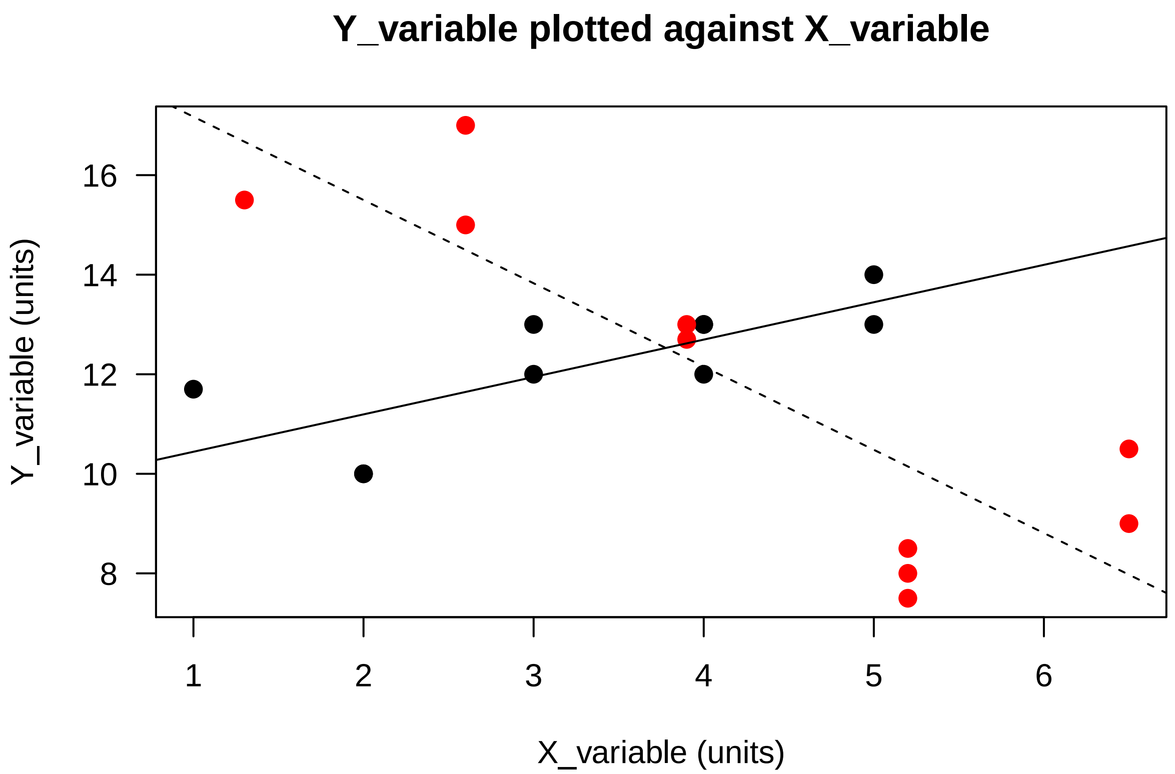
pch = shapes within the plot
cex = size of the shapes
col = color
las = turns the Y-axis numbers 90 degrees
abline = plots the best-fit line through the points (usually you DO NOT plot a line if there is no significant relationship, but you still show the points to visualize the spread in the data)

A quick google search can also help you customize your graphs further

Now what if you want to plot an ANCOVA (i.e., show two different lines through 2 sets of points related to each treatmnet). You could run the following code:

```
# First, make individual objects for your X and Y axis
X_variable <- Reg_data$X_variable
Y_variable <- Reg_data$Y_variable

# Run all lines of code together
plot(X_variable, Y_variable, pch=16, col = as.numeric(Reg_data$Treatment), cex = 1.3, las = 1, main = "
abline(lm(Y_variable[Reg_data$Treatment == 'Control'] ~ X_variable[Reg_data$Treatment == 'Control']), l
abline(lm(Y_variable[Reg_data$Treatment == 'Manipulation'] ~ X_variable[Reg_data$Treatment == 'Manipula
```



Be sure to describe in the figure legend which colors/line types correspond to which treatments.